Group 10: Paper Critical Review

My group's project statement is to develop a handheld prototype, consisting of a projector and depth camera, that can project patient data (e.g. CT/MRI scan) fixed onto patient body in real-time during surgeries.

For my paper critical review, I have chosen a paper that I believe provides a great foundation for my project. The paper is called "PMOMO: Projection Mapping on Moveable 3D Object" written by a lab from Shanghai Jiao Tong University and published in CHI 2016. This paper is crucial to our project because it illustrates an implementation of the core concept of projection mapping for a different application. Although the hardware that is used in this paper is not exactly the same as what is used in my project, the workflow pipeline and and final product is something similar to what I am aiming for.

The problem addressed in "PMOMO" is that realistic real-time projection mapping on dynamic objects is difficult to achieve. The accuracy in the projection is difficult to maintain when the object is manipulated at an interactive level, and the errors can accumulate over time. In addition, when the user occludes parts of the object, the projection mapping needs to realistically reflect those occlusions. The important result of this research is that the lab has developed a full projection mapping system that can align its projections in real-time with an object in 6DOF motion, support objects of arbitrary geometry, and remove occluded parts of the object from the texture being projected. The significance of this result is that allowing such a wide range of geometric complexity and motion for the target object can lead to a larger variety of AR applications in fields, such as art, education, entertainment, medicine, etc. It allows this

technology to be applied to more realistic situations where object complexity and motion can be arbitrary and unpredictable and occlusions of all shapes and sizes can occur.

Previous work has explored projection mapping on dynamic objects but has been restricted due to limited range or speed in object motion or limited occlusions. And for the hardware, some setups are simply too complicated while producing inaccurate projections. Tracking sensors, such as magnetic sensors, high-speed vision sensors, and optical markers, have been used, but unfortunately they present limited range in translation or rotation or interfere with the projection itself when placed on the target object. In terms of software, it is difficult to segment the desired object in real-time, especially in situations where the object is being held by the user or overlapped very closely by other objects. But for finding the transformation of the texture to the real-world object, the authors find that one promising algorithm is CMA-ES (Covariance Matrix Adaptation Evolution Strategy), which is an iterative optimization procedure, and it can reliably support object deformation.

The procedure of this research can be split into two phases: a preparation and real-time phase. In the software side of the preparation phase, a 3D model of the target object needs to be constructed, and in my project, that would be the equivalent of reconstructing a 3D model from CT scans. From this mesh model, four other models need to be generated. The first is a low-density point cloud used for tracking. The second is a high-density point cloud for dealing with occlusions. Both are evenly-distributed point clouds using Poisson-Disk sampling, and during this process, a point-facet list is also generated, which gives information about which facets are adjacent to each point. The third model is remeshed from the second model, and the

last model is a model with the desired texture for projection. The usage of these models will be explained in the real-time portion.

In the hardware side of the preparation phase, the team uses a Kinect 2.0, an AHRS sensor, and a projector, and there is calibration needed to be done for these various sensors. The first important step is that a virtual scene needs to be precisely calibrated to the real scene. The 3D model in the scene should be in the same pose and position relative to the camera as in the real world. The Kinect is used to capture the position of the target object while the AHRS sensor is used to capture the object's rotation. The target object in the scene is then updated with these values, and the virtual camera determines the projection content. For calibrating these sensors, a typical checkerboard calibration is done for the RGB and IR cameras. The RGB camera and projector are then calibrated using structured light patterns.

For the real-time phase, the first crucial part is tracking in which they use a modified form of CMA-ES. Every frame uses the transformation matrix found in the previous frame to calculate the new transformation that best registers the low-density point cloud to the corresponding depth image. The translation is obtained from the CMA-ES method while the rotation of the transformation matrix is replaced with the rotation obtained from the AHRS sensor. As the name suggests, the covariance matrix is used to generate some variants of the translation vectors, and then the point cloud is transformed according to each translation and the rotation obtained from the AHRS sensor. The visible points in the point cloud are selected, and the fitness value uses those visible points to determine the RMS distance of the transformed points in 3D space and their corresponding points in the depth image. The set of occluded points are then updated based on those transformed points. One drawback with the unadjusted CMA-ES

method is that it may fall into incorrect local optima, and in order to combat that, the team uses an adaptive step-size that grows smaller if the distance between two vectors of consecutive frames is more than a threshold. Because of this, the difference in positions of two point clouds of consecutive frames is limited. Furthermore, in order to further reduce any error, the team needed to deal with the different input delays of the multiple sensors. The AHRS sensor apparently has a negligible delay, but the Kinect has an approximately 60 ms delay. Therefore, the translation vector used for projection is calculated using a linear model while the rotation matrix can simply be derived from the most recent rotation of the AHRS sensor.

The second part of the real-time phase is dealing with occlusions. The team uses the point-facet list and the set of occluded points obtained during the tracking procedure. Because the points that are occluded are known, the point-facet list can then be used to determine which facets correspond to those occluded points. Then, these facets can be covered in the textured 3D model as the background color.

The team then conducts an experiment in order to determine the accuracy of their developed procedure. They had 10 volunteers freely move three different object models within the field of view of the Kinect camera and recorded around 100,000 frames. They calculated the average RMS Euclidean distance between the model and the corresponding depth image and categorized their frames into multiple categories based on the velocity and acceleration of the target object and the percent of the object occluded. They then compare their registration results and their linear model prediction results with the Kinect Fusion results. The results show that the translation prediction does not work well when the acceleration of the object is high, and at high

velocities with the occlusion percentage greater than 15%, the projection error grows very high, but the tracking is still reliable.

This paper provides a great basis for projection mapping technologies and thoroughly explains the technical setup and procedure to recreate the results. The adaptive aspects of the project, such as the occlusion adaptive threshold and step-size, allow there to be fewer hyperparameters, and the results become less application-specific. There is even a short analysis of the computation complexity of the system, which is something I have not encountered very much in other related research papers. In addition, the results are thorough and provide great comparisons to existing technologies, such as Kinect Fusion. However, I think there are a few extra steps the students could have taken. I wonder why the team chose a linear model for the predicted translation or if there are other models, which may be more accurate. In addition, I would have loved for an in-depth analysis of why these hardware systems were chosen or if there are comparable alternatives, especially if they are cheaper or more accurate without a large trade off. Furthermore, the table of results is categorized confusingly based on a combination of factors: range of acceleration, range of occlusion, and range of velocity. One drawback I see for this project is that there is a large amount of setup that needs to be done, especially with the model generation and the creation of an accurate virtual scene.

References

1. Yi Zhou, Shuangjiu Xiao, Ning Tang, Zhiyong Wei, and Xu Chen. 2016. **Pmomo: Projection Mapping on Movable 3D Object**. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 781-790.* DOI: https://doi.org/10.1145/2858036.2858329